

## AMENDMENTS TO CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

### **Listing of Claims:**

1. (Currently Amended) An XML document editor stored on a computer-readable medium to enable a user to add or delete an element into a working document having two consecutive elements  $z_i$  and  $z_{i+1}$  and to convert said working document into an XML document file $[[;]]$ , a relation between said elements  $z_i$  and  $z_{i+1}$  complying with a document type definition (DTD) of said document, characterized in that said XML document editor automatically generates, in relating relation to said two consecutive elements  $z_i$  and  $z_{i+1}$  of said working document, where relation between said elements  $z_i$  and  $z_{i+1}$  complies with document type definition (DTD) of said document, a list of candidate third element elements  $z$  to be alerted provided to a user; wherein said third element elements  $z$  in said list makes are selected such that relations between elements  $z_i$  and  $z$  and between elements  $z$  and  $z_{i+1}$  complying comply with said DTD $[[,]]$  after said element  $z$  is inserted between elements  $z_i$  and  $z_{i+1}$ , said list enabling the user to select any of the third elements and add them to the working document without affecting DTD compliance of the working document.

2. (Currently Amended) The XML document editor according to claim 1, wherein said XML document editor determines whether the relation between two consecutive elements comply complies with said DTD according to the following rule:

suppose  $G$  is Glushkov Automaton of said document,  $z_i$  is a state in  $G$ ,  $1 \leq i \leq p-1$ ,  $p \in \mathbb{N}$ ,  $\Sigma = \{z_1, z_2, \dots, z_p\}$  is a sequence of states in  $G$  where  $z_1 = s$ ,  $s$  is start state of  $G$ ,  $z_p = f$ ,  $f$  is final state of  $G$ ;

if  $z_{i+1} \in \text{reachable}(z_i)$ , wherein  $\text{reachable}(z_i)$  denote the set of states in  $G$  reachable from state  $z_i$ ,

then the relation between  $z_i$  and  $z_{i+1}$  is determined compliant with the DTD of said document.

3. (Currently Amended) The XML document editor according to claim 1, wherein said XML generates a cell C to include said candidate third element-elements z according to the following rule and displays said candidates in a list:

suppose  $(z_i, z_{i+1}) \in H$ , H denotes the set of edges in G, G is Glushkov Automaton of regular expression E corresponding to an element of said working document;

further suppose  $\Sigma$  is a set to include states corresponding to all elements of G,  $A(E1)$  is the set of states in subexpression E1 to E,  $f\text{-reachable}(z_i)$  denotes the set of states in G reachable from  $z_i$  through forward edges;

if  $z_{i+1} \in f\text{-reachable}(z_i)$ , then let  $C = \{ z \in \Sigma \mid z \in f\text{-reachable}(z_i) \text{ and } z_{i+1} \in f\text{-reachable}(z) \}$ ;

if  $z_{i+1} \notin f\text{-reachable}(z_i)$ , then let  $E1^*$  be the smallest iteration subexpression of E that covers both  $z_i$  and  $z_{i+1}$ ,  $C = \{ z \in A(E1) \mid z \in f\text{-reachable}(z_i) \text{ or } z_{i+1} \in f\text{-reachable}(z) \}$ .

4. (Currently Amended) The XML document editor according to claim 1, wherein said XML generates a cell C to include said candidate third element-elements z according to the following rule and displays said candidates in a list:

suppose  $(z_i, z_{i+1}) \in H$ , H denotes the set of edges in G, G is Glushkov Automaton of regular expression E corresponding to an element of said working document;

further suppose  $\Sigma$  is a set to include states corresponding to all elements of G,  $A(E1)$  is the set of states in subexpression E1 to E,  $f\text{-reachable}(z_i)$  denotes the set of states in G reachable from  $z_i$  through forward edges;

if  $(z_i, z_{i+1})$  is a forward edge, let  $C = \{ z \in \Sigma \mid z \in f\text{-reachable}(z_i) \text{ and } z_{i+1} \in f\text{-reachable}(z) \}$  and:

if  $z_i \in \text{last}(E1^*)$  for some iteration subexpression  $E1^*$  of E and  $E1$  is the largest one, then let  $C1 = \{ z \in A(E1) \mid z_{i+1} \in f\text{-reachable}(z) \}$ ,  $C = C \cup C1$ ;

If  $z_{i+1} \in \text{first}(E2^*)$  for some iteration subexpression  $E2^*$  of E and  $E2$  is the largest one, then let  $C2 = \{ z \in A(E2) \mid z \in f\text{-reachable}(z_i) \}$  and  $C = C \cup C2$ ;

if  $(z_i, z_{i+1})$  is a backward edge, then let  $C = A(E3)$ , wherein  $E3^*$  is the largest iteration subexpression of  $E$  satisfying  $z_i \in \text{last}(E3)$  and  $z_{i+1} \in \text{first}(E3)$ .

5. (Original) The XML document editor according to claim 1, wherein said XML document editor automatically generates a required element between element pair  $z_i$  and  $z$  and element pair  $z$  and  $z_{i+1}$  after said third element  $z$  is inserted between element pair  $z_i$  and  $z_{i+1}$ , such that said working document is effective; wherein said requirement comprises articulation points between elements  $z_i$  and  $z$  (and  $z$  and  $z_{i+1}$ ) in Glushkov Automaton  $G$ ; i.e., states through which all paths between  $z_i$  and  $z$  (and  $z$  and  $z_{i+1}$ ) shall pass.

6. (Original) The XML document editor according to claim 1, wherein said XML document editor automatically generates an element slot allowing user to add elements into said document, if no required element between element pair  $z_i$  and  $z$  and element pair  $z$  and  $z_{i+1}$  is found after said third element  $z$  is inserted between element pair  $z_i$  and  $z_{i+1}$  and if  $(z_i, z) \in H \cap (z, z_{i+1}) \in H$ , wherein  $H$  denotes set of edges in  $G$ ; and wherein said requirement comprises articulation points between elements  $z_i$  and  $z$  (and  $z$  and  $z_{i+1}$ ) in Glushkov Automaton  $G$ ; i.e., states through which all paths between  $z_i$  and  $z$  (and  $z$  and  $z_{i+1}$ ) shall pass.

7. (Currently Amended) Method for editing enabling a user to more easily edit an XML document using an XML document editor to enable the user to add or delete an element into a working document and to convert said working document into an XML document file; characterized in that said method comprising the step of enabling said XML document editor to automatically generate, in relation to two consecutive elements  $z_i$  and  $z_{i+1}$  of said working document, wherein relation between said elements  $z_i$  and  $z_{i+1}$  complies with document type definition (DTD) of said document, a list of candidate third element elements  $z$  to be alerted provided to a user; wherein said third element elements  $z$  in said list makes are selected such that relations between elements  $z_i$  and  $z$  and between elements  $z$  and  $z_{i+1}$  complying comply with said DTD[[,]] after said element  $z$  is inserted between elements  $z_i$  and  $z_{i+1}$ , said list enabling the user

to select any of the third elements and add them to the working document without affecting DTD compliance of the working document

8. (Currently Amended) The method according to claim 7, wherein whether the relation between two consecutive elements complies with said DTD is determined according to the following rule:

suppose G is Glushkov Automaton of said document,  $z_i$  is a state in G,  $1 \leq i \leq p-1$ ,  $p \in \mathbb{N}$ ,  $\Sigma = \{z_1, z_2, \dots, z_p\}$  is a sequence of states in G where  $z_1 = s$ , s is start state of G,  $z_p = f$ , f is final state of G;

if  $z_{i+1} \in \text{reachable}(z_i)$ , wherein  $\text{reachable}(z_i)$  denote the set of states in G reachable from state  $z_i$ ,

then the relation between  $z_i$  and  $z_{i+1}$  is determined compliant with DTD of said document.

9. (Currently Amended) The method according to claim 7, wherein a cell C to include said candidate third elementelements z is generated according to the following rule and displayed as a list:

suppose  $(z_i, z_{i+1}) \in H$ , H denotes the set of edges in G, G is Glushkov Automaton of regular expression E corresponding to an element of said working document;

further suppose  $\Sigma$  is a set to include states corresponding to all elements of G,  $A(E1)$  is the set of states in subexpression E1 to E,  $f\text{-reachable}(z_i)$  denotes the set of states in G reachable from  $z_i$  through forward edges;

if  $z_{i+1} \in f\text{-reachable}(z_i)$ , then let  $C = \{z \in \Sigma \mid z \in f\text{-reachable}(z_i) \text{ and } z_{i+1} \in f\text{-reachable}(z)\}$ ;

if  $z_{i+1} \notin f\text{-reachable}(z_i)$ , then let  $E1^*$  be the smallest iteration subexpression of E that covers both  $z_i$  and  $z_{i+1}$ ,  $C = \{z \in A(E1) \mid z \in f\text{-reachable}(z_i) \text{ or } z_{i+1} \in f\text{-reachable}(z)\}$ .

10. (Currently Amended) The method according to claim 7, wherein a cell C to include said candidate third elementelements z is generated according to the following rule and displayed as a list:

suppose  $(z_i, z_{i+1}) \in H$ , H denotes the set of edges in G, G is Glushkov Automaton of regular expression E corresponding to an element of said working document;

further suppose  $\Sigma$  is a set to include states corresponding to all elements of  $G$ ,  $A(E1)$  is the set of states in subexpression  $E1$  to  $E$ ,  $f\text{-reachable}(z_i)$  denotes the set of states in  $G$  reachable from  $z_i$  through forward edges;

if  $(z_i, z_{i+1})$  is a forward edge, let  $C = \{ z \in \Sigma \mid z \in f\text{-reachable}(z_i) \text{ and } z_{i+1} \in f\text{-reachable}(z) \}$  and:

if  $z_i \in \text{last}(E1^*)$  for some iteration subexpression  $E1^*$  of  $E$  and  $E1$  is the largest one, then let  $C1 = \{ z \in A(E1) \mid z_{i+1} \in f\text{-reachable}(z) \}$ ,  $C = C \cup C1$ ;

If if  $z_{i+1} \in \text{first}(E2^*)$  for some iteration subexpression  $E2^*$  of  $E$  and  $E2$  is the largest one, then let  $C2 = \{ z \in A(E2) \mid z \in f\text{-reachable}(z_i) \}$  and  $C = C \cup C2$ ;

if  $(z_i, z_{i+1})$  is a backward edge, then let  $C = A(E3)$ , wherein  $E3^*$  is the largest iteration subexpression of  $E$  satisfying  $z_i \in \text{last}(E3)$  and  $z_{i+1} \in \text{first}(E3)$ .

11. (Original) The method according to claim 7, further comprising automatically generating a required element between element pair  $z_i$  and  $z$  and element pair  $z$  and  $z_{i+1}$  after said third element  $z$  is inserted between element pair  $z_i$  and  $z_{i+1}$ , such that said working document is effective; wherein said requirement comprises articulation points between elements  $z_i$  and  $z$  (and  $z$  and  $z_{i+1}$ ) in Glushkov Automaton  $G$ ; i.e., states through which all paths between  $z_i$  and  $z$  (and  $z$  and  $z_{i+1}$ ) shall pass.

12. (Original) The method according to claim 7, further comprising automatically generating an element slot allowing user to add elements into said document, if no required element between element pair  $z_i$  and  $z$  and element pair  $z$  and  $z_{i+1}$  is found after said third element  $z$  is inserted between element pair  $z_i$  and  $z_{i+1}$  and if  $(z_i, z) \in H \cup (z, z_{i+1}) \in H$ , wherein  $H$  denotes set of edges in  $G$ ; and wherein said requirement comprises articulation points between elements  $z_i$  and  $z$  (and  $z$  and  $z_{i+1}$ ) in Glushkov Automaton  $G$ ; i.e., states through which all paths between  $z_i$  and  $z$  (and  $z$  and  $z_{i+1}$ ) shall pass.

13. (Original) An XML document editor stored on a computer-readable medium, comprising a user interface enabling user to add or delete an element into a working document, whereby said

working document is converted into an XML document file; characterized in that said XML document editor automatically generates in relating relation to two consecutive elements  $z_i$  and  $z_{i+1}$  of said working document, wherein relation between said elements  $z_i$  and  $z_{i+1}$  complies with document type definition (DTD) of said document, a list of candidate third element elements  $z$  to be alerted provided to a user; wherein said third element elements  $z$  in said list makes are selected such that relations between elements  $z_i$  and  $z$  and between elements  $z$  and  $z_{i+1}$  complying comply with said DTD[[,]] after said element  $z$  is inserted between elements  $z_i$  and  $z_{i+1}$ , said list enabling the user to select any of the third elements and add them to the working document without affecting DTD compliance of the working document

14. (Currently Amended) The XML document editor according to claim 13, wherein said XML document editor determines whether the relation between two consecutive elements comply with said DTD according to the following rule:

suppose  $G$  is Glushkov Automaton of said document,  $z_i$  is a state in  $G$ ,  $1 \leq i \leq p-1$ ,  $p \in \mathbb{N}$ ,  $\Sigma = \{z_1, z_2, \dots, z_p\}$  is a sequence of states in  $G$  where  $z_1 = s$ ,  $s$  is start state of  $G$ ,  $z_p = f$ ,  $f$  is final state of  $G$ ;

if  $z_{i+1} \in \text{reachable}(z_i)$ , wherein  $\text{reachable}(z_i)$  denote the set of states in  $G$  reachable from state  $z_i$ ,

then the relation between  $z_i$  and  $z_{i+1}$  is determined compliant with DTD of said document.

15. (Currently Amended) The XML document editor according to claim 13, wherein said XML generates a cell  $C$  to include said candidate third element elements  $z$  according to the following rule and displays said candidates in a list:

suppose  $(z_i, z_{i+1}) \in H$ ,  $H$  denotes the set of edges in  $G$ ,  $G$  is Glushkov Automaton of regular expression  $E$  corresponding to an element of said working document;

further suppose  $\Sigma$  is a set to include states corresponding to all elements of  $G$ ,  $A(E1)$  is the set of states in subexpression  $E1$  to  $E$ ,  $f\text{-reachable}(z_i)$  denotes the set of states in  $G$  reachable from  $z_i$  through forward edges;

if  $z_{i+1} \in f\text{-reachable}(z_i)$ , then let  $C = \{ z \in \Sigma \mid z \in f\text{-reachable}(z_i) \text{ and } z_{i+1} \in f\text{-reachable}(z) \}$ ;

if  $z_{i+1} \in f\text{-reachable}(z_i)$ , then let  $E1^*$  be the smallest iteration subexpression of  $E$  that covers both  $z_i$  and  $z_{i+1}$ ,  $C = \{z \in A(E1) \mid z \in f\text{-reachable}(z_i) \text{ or } z_{i+1} \in f\text{-reachable}(z)\}$ .

16. (Currently Amended) The XML document editor according to claim 13, wherein said XML generates a cell  $C$  to include said candidate third elementelements  $z$  according to the following rule and displays said candidates in a list:

suppose  $(z_i, z_{i+1}) \in H$ ,  $H$  denotes the set of edges in  $G$ ,  $G$  is Glushkov Automaton of regular expression  $E$  corresponding to an element of said working document;

further suppose  $\Sigma$  is a set to include states corresponding to all elements of  $G$ ,  $A(E1)$  is the set of states in subexpression  $E1$  to  $E$ ,  $f\text{-reachable}(z_i)$  denotes the set of states in  $G$  reachable from  $z_i$  through forward edges;

if  $(z_i, z_{i+1})$  is a forward edge, let  $C = \{z \in \Sigma \mid z \in f\text{-reachable}(z_i) \text{ and } z_{i+1} \in f\text{-reachable}(z)\}$  and:

if  $z_i \in \text{last}(E1^*)$  for some iteration subexpression  $E1^*$  of  $E$  and  $E1$  is the largest one, then let  $C1 = \{z \in A(E1) \mid z_{i+1} \in f\text{-reachable}(z)\}$ ,  $C = C \cup C1$ ;

If if  $z_{i+1} \in \text{first}(E2^*)$  for some iteration subexpression  $E2^*$  of  $E$  and  $E2$  is the largest one, then let  $C2 = \{z \in A(E2) \mid z \in f\text{-reachable}(z_i)\}$  and  $C = C \cup C2$ ;

if  $(z_i, z_{i+1})$  is a backward edge, then let  $C = A(E3)$ , wherein  $E3^*$  is the largest iteration subexpression of  $E$  satisfying  $z_i \in \text{last}(E3)$  and  $z_{i+1} \in \text{first}(E3)$ .

17. (Original) The XML document editor according to claim 13, wherein said XML document editor automatically generates a required element between element pair  $z_i$  and  $z$  and element pair  $z$  and  $z_{i+1}$  after said third element  $z$  is inserted between element pair  $z_i$  and  $z_{i+1}$ , such that said working document is effective; wherein said requirement comprises articulation points between elements  $z_i$  and  $z$  (and  $z$  and  $z_{i+1}$ ) in Glushkov Automaton  $G$ ; i.e., states through which all paths between  $z_i$  and  $z$  (and  $z$  and  $z_{i+1}$ ) shall pass.

18. (Original) The XML document editor according to claim 13, wherein said XML document editor automatically generates an element slot allowing user to add elements into said document,

if no required element between element pair  $z_i$  and  $z$  and element pair  $z$  and  $z_{i+1}$  is found after said third element  $z$  is inserted between element pair  $z_i$  and  $z_{i+1}$  and if  $(z_i, z) \in H$   $((z, z_{i+1}) \in H)$ , wherein  $H$  denotes set of edges in  $G$ ; and wherein said requirement comprises articulation points between elements  $z_i$  and  $z$  (and  $z$  and  $z_{i+1}$ ) in Glushkov Automaton  $G$ ; i.e., states through which all paths between  $z_i$  and  $z$  (and  $z$  and  $z_{i+1}$ ) shall pass.